



## Module : Algorithmique et Structures de Données 1 (ASD1)

### Chapitre 5 : Les chaines de caractères

#### 1. Les chaines de caractères

##### 1.1. Définition

Les chaînes de caractères sont des séquences finies de caractères. Le nombre de caractères composant une chaîne de caractères est **la longueur** de cette chaîne.

Une chaîne de **longueur nulle** ne comprend aucun caractère : c'est **la chaîne vide**.

En algorithmique, une chaîne de caractères est désignée entre **deux apostrophes**.

##### Exemple :

- La chaîne de caractères '**Bonjour**' est constituée des caractères 'B', 'o', 'n', 'j', 'o', 'u' et 'r'.
- La chaîne de caractères '**123**' n'est constituée que de chiffres (à ne pas confondre avec le type numérique 123).
- La chaîne de caractères " représente une chaîne de caractères vide.
- La chaîne de caractères ' ' est une chaîne de caractères ne contenant qu'un seul caractère qui est ici le caractère espace (à ne pas confondre avec la chaîne de caractères vide).

##### 1.2. Déclaration

On déclare une chaîne de caractères en utilisant la syntaxe suivante :

- **En algorithmique** : On utilise deux façons en indiquant ou non le nombre de caractères :

**Var nom\_variable : chaine [longueur] ;**

**Ou bien**

**Var nom\_variable : chaine ;**

- **En langage C** :

Il n'existe pas de type spécial pour les chaînes de caractères en langage C. Une chaîne de caractères est déclarée comme un tableau à une dimension de char (vecteur de caractères) dont la fin est indiquée par le caractère '\0'. La taille de la chaîne est égale à la longueur maximale de la chaîne plus un pour qu'on puisse stocker le caractère '\0' dénotant la fin.

**char nom\_variable [longueur] ;**

**Exemple :**

- **En algorithmique**

**var** ch : chaîne [10] ; ou **var** ch : chaîne ;

- **En langage c**

**char** ch [10]; // la longueur maximale de la chaîne est 9

**1.3. Accès aux éléments d'une chaîne de caractères**

Pour accéder à l'**ième** élément d'une chaîne, il suffit de donner **l'identificateur de la chaîne** suivi de **l'indice i** entre crochets (comme un tableau).

**Exemple :**

- **En algorithmique :**

Ch← ‘Bonjour’; CH[1] = ‘B’; CH[3] = ‘n’; CH[6] = ‘u’;

- **En langage c :**

char CH[] = 'bonjour' ; CH[0] = ‘B’; CH[3] = ‘j’; CH[6] = ‘r’;

**1.4. Lecture et affichage**

En algorithmique, une chaîne de caractères est lue (affichée) globalement (d'un seul coup) et non pas caractère par caractère.

**Exemple :**

- **En algorithmique :**

Lire (CH) ; Ecrire (CH) ;

- **En langage C :** on peut lire ou afficher une chaîne de caractères de deux façons :

*Première façon* : **scanf** ("%s", CH) ; **printf** ("%s", CH) ;

*Deuxième façon* : **gets** (CH); **puts** (CH);

**1.5. Fonctions et procédures sur les chaînes de caractères**

Les fonctions et les procédures standards qui manipulent les chaînes de caractères sont nombreuses. Ici nous allons donner quelques d'eux.

- **En algorithmique**

## 1) Les fonctions standards relatives aux chaînes de caractères

<i>Syntaxe</i>	<i>Rôle</i>	<i>Type</i>		<i>Exemples</i>
		<i>Entrée</i>	<i>Sortie</i>	
<b>long</b> (ch)	Retourne un entier représentant la longueur de ch (nombre de caractères de ch)	Chaîne	Entier	L1 ← long ('Bonjour') ; // L=7 L2 ← long ("") ; // L=0 ; L3 ← long (' ') ; // L= 1 ;
<b>pos</b> (chl, ch2)	Retourne la première position de la chaîne ch1 dans la chaîne ch2.	Chaînes	Entier	P1← pos ('jour', 'Bonjour') ;// P=4 P2 ← pos ('o', 'Bonjour') ; // P= 2 P3 ← pos ('s', 'Bonjour') ;// P= 0
<b>copier</b> (ch, p, n)	Retourne une sous chaîne de n caractères à partir de la position p de la chaîne ch.	Chaîne, Entier, Entier	Chaîne	CH1 ← Copier ('Bonjour', 1, 3) ; // CH1 = 'Bon' CH2 ← Copier ('Bonjour', 4, 4) ; // CH2 = 'jour'
<b>concat</b> (chl, ch2, ...)	Retourne la concaténation (l'enchainement) de plusieurs chaînes en une seule. C'est l'équivalent de ch1+ch2+...	Chaînes	Chaîne	CH1 ← 'Juillet' ; CH2 ← Concat ('5', '/', CH1, '/', '1962') ; CH2 = '5/juillet/1962'

## 2) Les procédures standards relatives aux chaînes de caractères

<i>Syntaxe</i>	<i>Rôle</i>	<i>Type</i>		<i>Exemples</i>
		<i>Entrée</i>	<i>Sortie</i>	
<b>efface</b> (ch, p, n)	Supprime n caractères de la chaîne <b>ch</b> à partir de la position <b>p</b> .	Chaîne, Entier, Entier	Chaîne	CH← 'Bonjour' ; efface (CH, 1, 3) ; // CH ='jour'
<b>insere</b> (ch1, ch2, p)	Insère la chaîne <b>ch1</b> dans la chaîne <b>ch2</b> à la position <b>p</b> .	Chaîne/Caractère, Chaîne, Entier	Chaîne	CH1 ← 'rit' ; CH2 ← 'algomique' ; insere (CH1, CH2, 5) ; // CH2 devient 'algorithme'
<b>convch</b> (n, ch)	Convertit une valeur numérique <b>n</b> (entier, réel) en une chaîne <b>ch</b> .	Numérique, chaîne	Chaîne	convch (2024, CH) ;// CH = '2024'
<b>valeur</b> (ch, n, err)	Convertit une chaîne de caractères <b>ch</b> en une valeur numérique <b>n</b> . De plus, elle fournit un code <b>d'erreur err</b> qui indique si l'opération s'est déroulée correctement.	Chaîne/Caractère, Numérique, Entier	Numérique, Entier	valeur ('2024', n, err) ; n = 2024 et err=0 valeur ('15H30', n, err) ; n = 0 et err = 3

- En langage C

### Les fonctions standards relatives aux chaînes de caractères

<i>Syntaxe</i>	<i>Bibliothèque</i>	<i>Rôle</i>	<i>Type</i>	
			<i>Paramètres de la fonction</i>	<i>Résultat</i>
<b>Strlen (ch)</b>	String.h	Permet de connaître la longueur de la chaîne ch.	ch : Chaîne	Entier
<b>Strcpy (ch1, ch2)</b>	string.h	Copie le contenu de la chaîne ch2 dans la chaîne ch1.	ch1 : Chaîne ch2 : Chaine	Chaîne
<b>Strcat (ch1, ch2)</b>	string.h	Permet de concaténer les contenus de deux chaînes de caractères (ajouter le contenu de la chaîne ch2 à celui de la chaîne ch1).	ch1 : Chaîne ch2 : Chaine	Chaîne
<b>Strcmp (ch1, ch2)</b>	string.h	Permet de comparer deux chaînes de caractères.	ch1 : Chaîne ch2 : Chaine	Valeur positive si ch1>ch2 Valeur nulle si ch1=ch2 Valeur négative si ch1<ch2
<b>Strchr (ch1, ch)</b>	string.h	Recherche la présence du caractère ch dans la chaîne ch1.	ch1 : Chaîne ch : Caractère	Adresse de la première occurrence du caractère s'il existe sinon NUL
<b>Strpbrk (ch1, ch2)</b>	string.h	Recherche la présence d'un des caractères de la chaîne ch2 dans la chaîne ch1.	ch1 : Chaîne ch2 : Chaine	Adresse du premier caractère rencontré sinon NUL
<b>Strstr (ch1, ch2)</b>	string.h	Recherche la présence de la chaîne ch2 dans la chaîne ch1.	ch1 : Chaîne ch2 : Chaine	Adresse de la première occurrence de la chaîne si elle existe sinon NUL
<b>Atoi (ch)</b>	stdlib.h	Convertit ch en entier	ch : Chaine	Entier
<b>Atol (ch)</b>	stdlib.h	Convertit ch en entier long	ch : Chaine	Entier long
<b>Atof (ch)</b>	stdlib.h	Convertit ch en double (réel)	ch : Chaine	Double
<b>strdel (ch,p,n)</b>	publib.h	Supprime n caractères de la chaîne ch à partir de la position p et stocke le résultat dans la chaîne ch.	ch : chaine p,n : entier	Chaine